Projekt

Sterowniki robotów

Raport

Humanistycznie upośledzony robot akrobatyczny

HURA

Skład grupy:
Albert Lis, 235534
Michał Moruń, 235986

Termin: sr TP15

Prowadzący:
mgr inż. Wojciech Domski

9 kwietnia 2019

Spis treści

1 Opis projektu **2**

2 Konfiguracja mikrokontrolera **2**
2.1 Konfiguracja pinów 5
2.2 ADC 1 5
2.3 Timer 2 5
2.4 Timer 4 6
2.5 Timer 6 6

3 Urządzenia zewnętrzne **6**

4 Projekt elektroniki **6**
4.1 Schemat elektryczny 6
4.2 Regulacja prędkości napędu 6

5 Konstrukcja mechaniczna **7**

6 Opis działania programu **7**
6.1 Schemat działania programu 7
6.2 Funkcja obsługująca przerwanie timera 6 7
6.3 Funkcja obsługująca przerwanie ADC 7

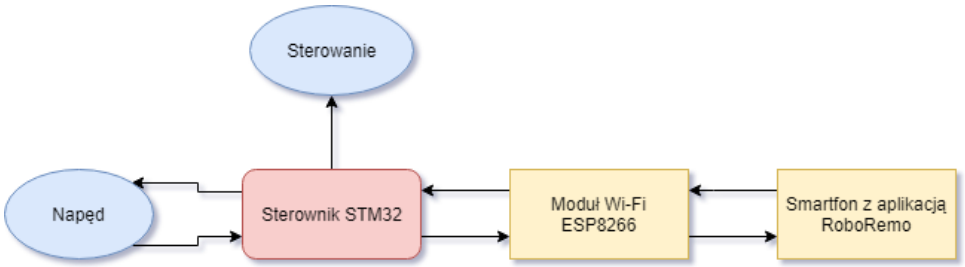
7 Zadania niezrealizowane **8**

8 Podsumowanie **8**

Bibilografia **9**

1 Opis projektu

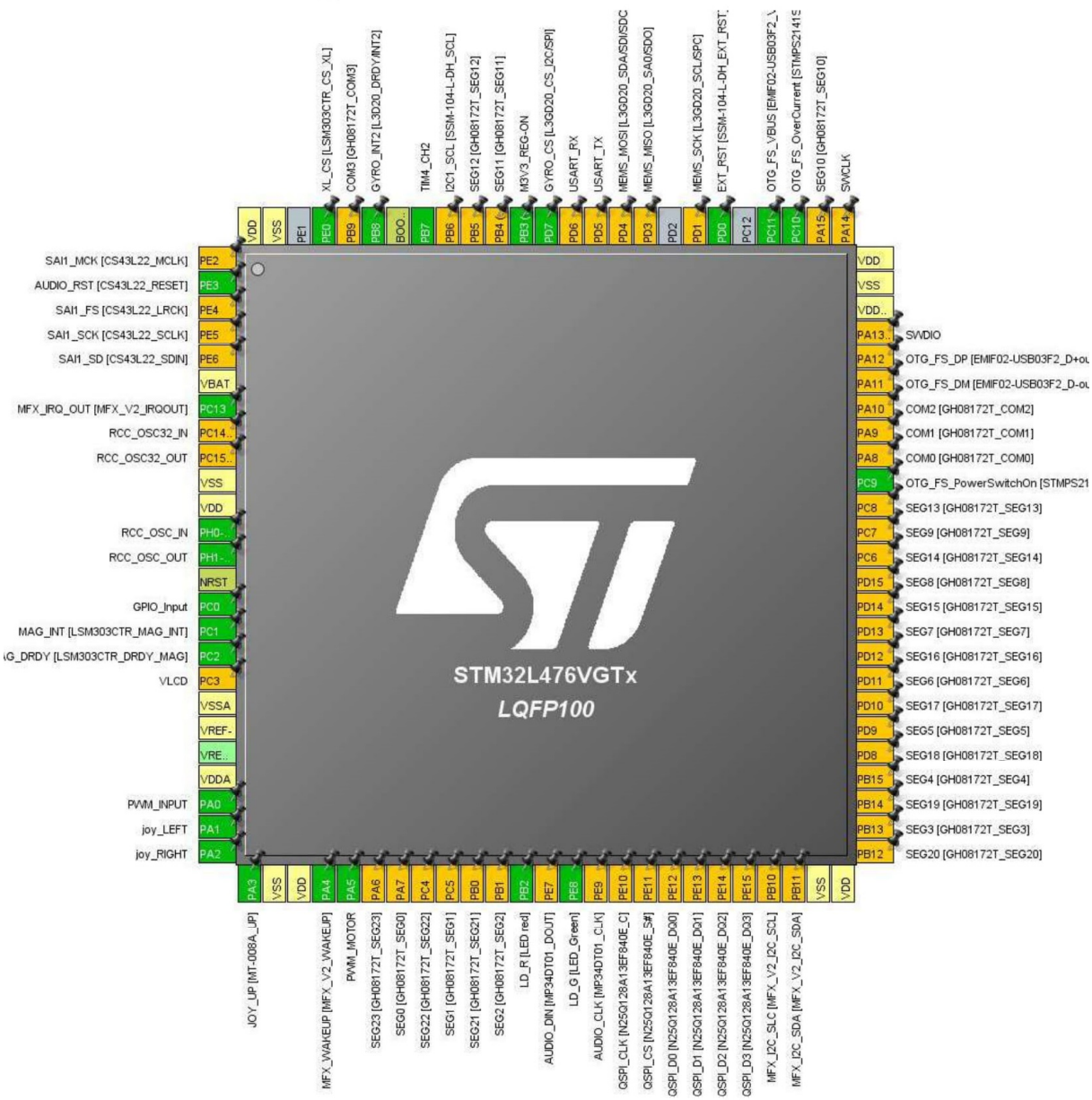
Celem projektu jest zbudowanie zdalnie sterowanego robota jezdnego. Robot będzie sterowany za pomocą akcelerometru w telefonie. Dane będą przesyłanie za pomocą Wi-Fi lub Bluetooth. Regulacja prędkości będzie się odbywać za pomocą regulatora PID. Dane o prędkości będą pobierane z enkoderów znajdujących się w kołach robota. Opcjonalnie robot będzie wyświetlał szczegółowe dane o swoim stanie wewnętrznym za pomocą wbudowanego w płytkę z mikrokontrolerem wyświetlacza LCD.



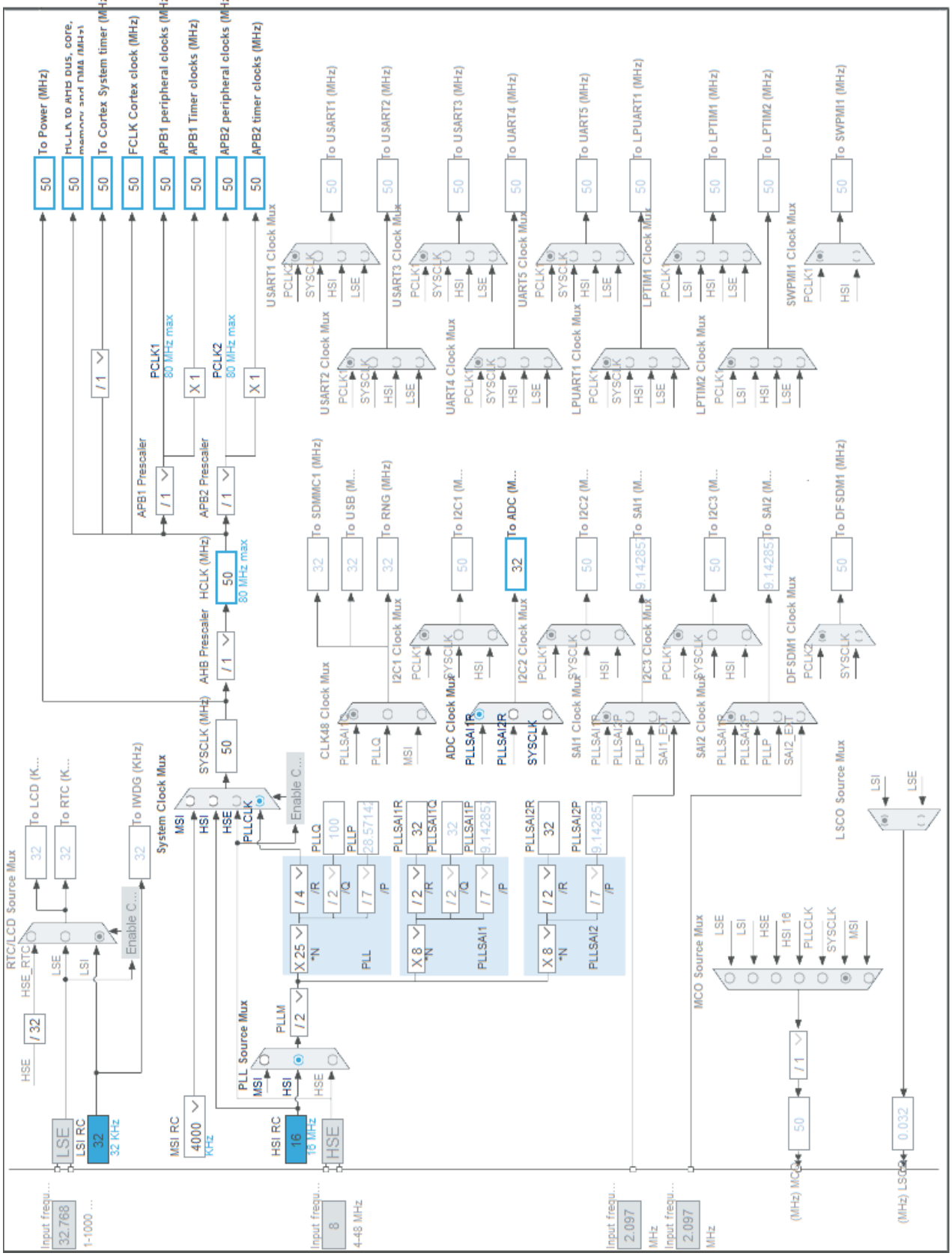
Rysunek 1: Architektura systemu

2 Konfiguracja mikrokontrolera

Tutaj powinna znaleźć się konfigurację poszczególnych peryferiów mikrokontrolera – jeśli wykorzystywany jest np. ADC to należy podać jego konfigurację nie zapominając o DMA jeśli jest wykorzystywane. Proszę wzorować się na raporcie wygenerowanym z programu STM32CubeMx (plik PDF i TXT, Project -> Generate Report Ctrl+R). W pliku PDF jest to rozdział *IPs* and *Middleware Configuration*. Należy umieścić uproszczoną konfiguracje peryferiów w formie tabelek (najistotniejsze parametry + parametry zmienione, pogrubione). Dodatkowo w pliku tekstowym (TXT) znajduje się konfiguracja pinów mikrokontrolera, którą również należy zamieścić w raporcie. W przypadku, gdy projekt zakłada wykorzystanie większej liczby modułów sekcję tą należy podzielić na odrębne podsekcje.



Rysunek 2: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX



Rysunek 3: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

PIN	Tryb pracy	Funkcja/etykieta
PC14	OSC32_IN* RCC_OSC32_IN	
PC15	OSC32_OUT* RCC_OSC32_OUT	
PH0	OSC_IN* RCC_OSC_IN	
PH1	OSC_OUT*	RCC_OSC_OUT
PD5	USART2_TX	USART_TX
PD6	USART2_RX	USART_RX
PA0	ADC1_IN5	PWM_INPUT
PA1	GPIO_Input	JOY_LEFT
PA2	GPIO_Input	JOY_RIGHT
PA3	GPIO_Input	JOY_UP
PA4	GPIO_Input	JOY_DOWN
PA5	TIM2_CH1	PWM_MOTOR
PB7	TIM4_CH2	PWM_SERVO

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 ADC 1

Parametr	Wartość
Resolution	ADC 12-bit resolution
DMA Continuous Requests	Enabled
Data Alignment	Right alignment
Continuous Conversion Mode	Disabled
Channel	Channel 5
Sampling Time	92.5 Cycles

Tabela 2: Konfiguracja peryferium ADC

2.3 Timer 2

Parametr	Wartość
Clock Source	Internal Clock
Channel1	PWM Generation CH1
Prescaler	PWM_PRESC
Counter Mode	Up
Counter Period	PWM_PERIOD
Internal Clock Division	No Division
Mode	PWM mode 1
CH Polarity	High

Tabela 3: Konfiguracja peryferium Timer 2

2.4 Timer 4

Parametr	Wartość
Clock Source	Internal Clock
Channel	PWM Generation CH2
Prescaler	999
Counter Mode	Up
Counter Period	999
Internal Clock Division	No Division
Mode	PWM mode 1
CH Polarity	High

Tabela 4: Konfiguracja peryferium Timer 4

2.5 Timer 6

Parametr	Wartość
Prescaler	TIM6_PRESC
Counter Mode	Up
Counter Period	TIM6_PERIOD
Trigger Event Selection	Update Event

Tabela 5: Konfiguracja peryferium Timer 6

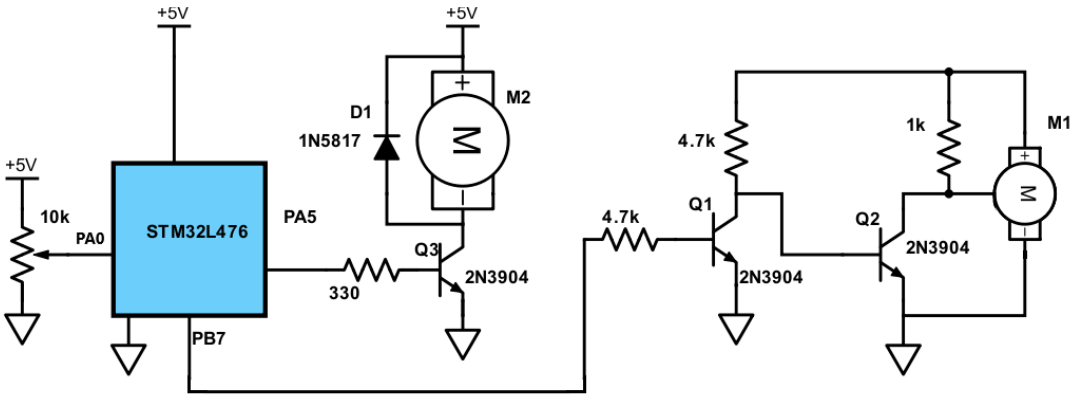
3 Urządzenia zewnętrzne

Rozdział ten powinien zawierać opis i konfigurację wykorzystanych układów zewnętrznych, jak np. akcelerometr.

????????

4 Projekt elektroniki

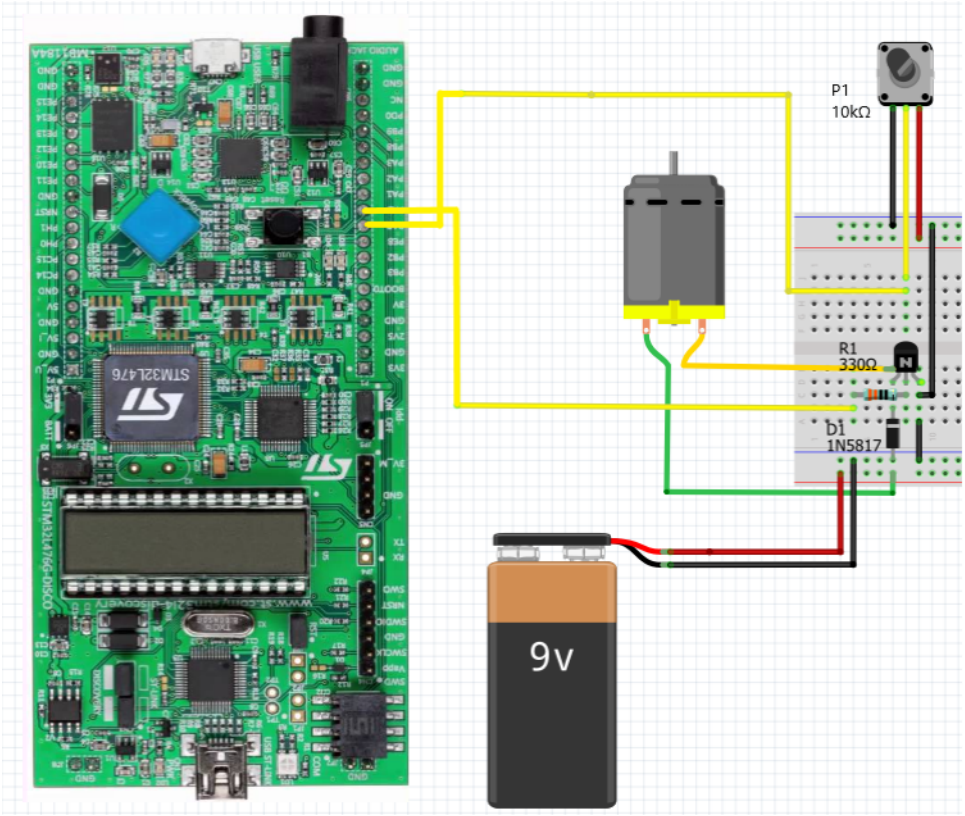
4.1 Schemat elektryczny



Rysunek 4: Schemat elektryczny

4.2 Regulacja prędkości napędu

Za pomocą potencjometru regulujemy wypełnienie sygnału PWM. Sygnał ten jest wzmacniany za pomocą tranzystora NPN i przekazywany do silnika DC.



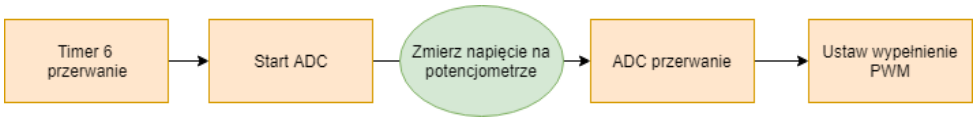
Rysunek 5: Schemat poglądowy regulacji prędkości obrotowej silnika

5 Konstrukcja mechaniczna

W przypadku, w którym projekt uwzględnia zastosowanie mechaniki to wówczas jej opis powinien znaleźć się tutaj. Nie należy dzielić rysunków mechaniki na poszczególne rzuty, wystarczy zamieścić wyrenderowane modele 3D. Można również dołączyć zdjęcia wykonanej mechaniki po uprzednim skompresowaniu, aby wynikowy rozmiar skompilowanego dokumentu nie był za duży.

6 Opis działania programu

6.1 Schemat działania programu



Rysunek 6: Schemat działania programu

6.2 Funkcja obsługująca przerwanie timera 6

```
1  void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
2  {
3    if(htim->Instance == TIM6)
4      HAL_ADC_Start_DMA(&hadc1, (uint32_t *)&adc_value, 1);
5  }
```

6.3 Funkcja obsługująca przerwanie ADC

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
  //pid_output = pid_calc(&pid, adc_value, set_value);
  __HAL_TIM_SET_COMPARE(&htim2 , TIM_CHANNEL_1, adc_value);
}
```

7 Zadania niezrealizowane

Jeśli wszystkie zadania zostały realizowane to wówczas ta sekcja powinna być usunięta w całości. W przeciwnym razie należy zawrzeć tutaj, jakie zadania zostały nie zrealizowane oraz jaka była tego przyczyna.

8 Podsumowanie

Krótkie podsumowanie projektu

Udało się zrealizować większość zadań. Nastąpiły drobne zmiany koncepcyjne jak użycie potencjometru do regulacji prędkości obrotowej napędu. To będzie wymagać mniejszej ingerencji gdy będziemy projektować regulator PID.

Literatura

[1] W. Domski. Sterowniki robotów, Laboratorium – Wprowadzenie, Wykorzystanie narzędzi STM32CubeMX oraz SW4STM32 do budowy programu mrugającej diody z obsługą przycisku. Mar. 2017.